([M|mac)?s?OS(sX)?: Permit ICMP redirects

So I was fighting around with that Motorola/Netopia router I'm obliged to use, because the network operator doesn't allow hooking up a custom device.

Well, it is possible after all, as proven in the past, however, in order to use the SIP gateway of network operator (whereas the login credentials are not provided), the operator-branded router must be used. *sigh*

Here's a somewhat high-level overview: The clients, which shall connect to the lab, are in the same subnet as the default router #1. The destination for more specific lab routes is router #2, which is in the same subnet.

```
+-----------+      +------------+      +-----------+      +-----------+      +-----------+
| clients   |  ----  | WiFi/Wired | ----  | router #1 | ----  | router #2 | ----  | LAB stuff |
+-----------+      +------------+      +-----------+      +-----------+      +-----------+
{                     CLIENT SUBNET                     } {     LAB SUBNETS     }
```

So actually, I could just add the more specific routes to any client, indicating it shall forward through router #2.
However, this is cumbersome. I wond't want to add these routes on every client.

So I tried hacking them into the Motorola/Netopia router. I had my hard time with that, but only because it's so silly on overly complicated ... :-(
So far so good, my clients could send ICMP echo requests towards the LAB devices, however, that was as close as I could get.

Not every client was capable in accessing everything in the LAB.

As it turned out, the Motorola/Netopia sends ICMP redirects. It does that because router #2 (a cisco, btw) is reachable via the CLIENT subnet and thus directly reachable by anyone in the same subnet.
However, ICMP redirects are somewhat non-deterministic, as the forwarding is not influenced by the router anymore. I consider it vodoo, which is why I prefer turning it off.

The only problem is that this "Netopia SOC OS" doesn't have an equivalent to a Cisco-type "no ip redirects"-command.
Well, it's a Linux after all, so I could turn it off by setting **/proc/sys/net/ipv4/conf/*/send_redirects** to **0**. There is an obscure way to break out from the SOC OS shell and get a unrestricted shell:
ping 127.0.0.2;/bin/busybox telnetd -l/bin/sh -p9999

This would open a root shell on port 9999, from where the kernel setting could be changed. However, since this will get reverted whenever the router reboots due to operator updates, I would need to hack this back in. I don't like this at all. Please, let me officially retrieve the SIP credentials to hook up my IP phone directly, so I can use a Cisco router. Pretty please!

Well, one day perhaps. Until then, I need to get it working with the least intrusive means of configuration.
So, I can't replace the router, I can't learn it to not send redirects.
But, if my clients, ([M|m]ac)?s?OS(sX)? in particular, don't play well with ICMP redirects, let's force them to do so.

On macOS (man, let's blatandly change the name one more time!), this can be done via the **sysctl** command in the Terminal.
Query it like this:

```
# sudo sysctl net.inet.icmp.drop_redirect
net.inet.icmp.drop_redirect: 1
```

So macOS indeed drops ICMP redirects by default. Let's change this:

```
# sudo sysctl net.inet.icmp.drop_redirect=0
net.inet.icmp.drop_redirect: 1 -> 0
```

With the new setting, connections started to work right away.

A note of caution: In my opinion ICMP redirects can be a dangerous thing, as they open the door for an attacker to influence the client's idea of the routing table.
A client should not listen to redirects and always forward traffic towards it's designated router. Overriding a default setting like this may be ok on a case-by-case basis, but should be strongly inspected und monitored.
If I had the choice, I surely had preferred a permanently applied setting on the router instead.